KOREAN PATENT ABSTRACTS

(11)Publication number: 1020000038953 A

(43)Publication date: 05.07.2000

(21)Application number: 1019980054131

(22)Application date: 10.12.1998

(51)Int. Cl H04L 1/00

(71)Applicant:
- SAMSUNG ELECTRONICS CO., LTD.

(72)Inventor:
- KIM, MIN GU

(54) INTERLEAVING/DEINTERLEAVING DEVICE IN COMMUNICATION SYSTEM AND METHOD FOR THE SAME

(57) Abstract:

PURPOSE: An interleaving/deinterleaving device in a communication system and a method for the same are provided to generate addresses through one algorithm for various sizes of interleaver in a communication system.

CONSTITUTION: An interleaving/deinterleaving device in a communication system and a method for the same comprises an address generation portion(211) and an interleaving memory(212). The address generation portion outputs values of interleaving addresses by inputting a size of an interleaver and values of variables according to the size of the interleaver. The interleaving memory stores input data and outputs data corresponding to address values provided from the address generation portion.

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.[6]
H04L 1/00

(11) 공개번호 특2000-0038953
(43) 공개일자 2000년07월05일

(21) 출원번호 10-1998-0054131
(22) 출원일자 1998년12월10일

(71) 출원인 삼성전자 주식회사 윤종용
경기도 수원시 팔달구 매탄3동 416
(72) 발명자 김민구
경기도 성남시 분당구 서현동 263
(74) 대리인 이건주

심사청구 : 있음

(54) 통신시스템의인터리빙/디인터리빙장치및방법

요약

통신시스템에서 인터리버 메모리에 순차적으로 저장된 데이타를 인터리빙하여 출력하기 위한 방법이, 인터리버 크기를 이진형태로 변환하여 최하위 비트로부터 최상위 비트 방향으로 연속되는 '0'비트수에 해당하는 영계수값을 구하는 단계와, 상기 '0'비트들을 제외한 비트들을 십진수로 변환하여 상한값을 구하는 단계와, 출력 순서번호를 상기 상한값으로 나누어 나머지값에 해당하는 제1변수값과 몫에 해당하는 제2변수값을 구하는 단계와, 상기 몫을 이진 형태로 변환한 후 비트역상순하여 십진형태로 변환하여 제3변수값을 구하는 단계와, 상기 나머지값과 2를 상기 영계수값으로 제곱근한 값을 곱한 후 이를 상기 제3변수값과 더해 인터리빙 어드레스 번지값을 생성하는 단계로 이루어져, 심볼 데이터가 순차적으로 저장된 인터리버 메모리에서 상기 생성된 인터리빙 어드레스 번지값에 대응되는 데이터를 출력한다.

대표도

도2

명세서

도면의 간단한 설명

도 1은 종래기술에 따른 비트역상순 인터리버의 치환방식을 설명한 도면.

도 2는 본 발명의 실시 예에 따른 인터리버의 구성도.

도 3은 본 발명의 실시 예에 따른 디인터리버의 구성도.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 통신시스템에 관한 것으로, 특히 무선통신시스템의 인터리빙/디인터리빙 장치 및 방법에 관한 것이다.

인터리빙은 페이딩 채널에서 오류정정부호의 성능을 향상시키기 위해서 사용하는 기술로 랜덤오류정정부호의 복호화와 밀접한 관계가 있다. 특히 최근에 매우 많은 관심을 모으고 있는 IMT-2000(CDMA2000)의 에어 인터페이스(air interface)에서 다양한 인터리빙 방식에 대한 구체적인 구현방식이 요구되고 있다. 또한, 이 분야는 디지털 통신시스템의 신뢰도 향상에 광범위하게 관련된 분야로서, 기존의 디지털 통신시스템의 성능개선 분야와 향후 결정되는 차세대 시스템의 성능을 개선시키는 방식에 관한 기술분야이기도 하다.

IMT-2000 분야에서는 채널 인터리버로 비트역상순 인터리버(bit reversal interleaver)를 사용하도록 잠정적으로 권고하고 있다. 그러나 IMT-2000 분야에에 주어진 순방향 링크(forward link)와 역방향 링크(reverse link)의 경우 논리채널(Logical channel)의 종류가 매우 다양하며, 인터리버의 크기 또한 여러 종류로 구성되어 이를 충실히 반영하기 위해서는 많은 양의 메모리가 요구된다. 예로서 순방향 링크 N=3(?)인 전송모드의 경우 최소 144bits/frame부터 최대 36864bits/frame 까지 매우 다양한 크기의 인터리버가 사용된다. 이러한 비트역상순 인터리버에 대한 개략적인 설명은 다음과 같다.

비트역상순 인터리버의 치환방식을 도 1에 보였다. 상기 도 1를 참조하면, 최상위 비트인 MSB(Most Significant Bit)로부터 최하위 비트인 LSB(Least Significant Bit)까지의 비트 위치(bit position)을 상

호 교환하여 재정렬시키는 방식을 비트역상순 인터리버라 한다. 이와 같은 방식의 장점은 열거(셈) 함수(enumeration function)을 사용하여 인터리버를 구현할수 있으므로 메모리 사용이 간단하며, 또한 여러 크기의 인터리버를 구현하기 쉽다. 또한 상기 치환된 시퀀스의 위치분포가 상당 부분 랜덤하게 이루어진다. 하지만 2의 거듭제곱 형태로 표현될수 없는 인터리버 크기의 경우 메모리 활용면에서 비효율적이라는 문제점을 가진다. 예를 들어, 36864비트의 인터리버의 경우 가장 쉽게 구현하기 위해서는 64k(65536=2^{16})의 메모리가 필요하다. 왜냐하면, 36864보다 큰 가장 작은 정수중 2의 거듭제곱으로 표현될수 있는 정수는 65536이기 때문이다. 따라서 28672(=65536-36864) 만큼의 사용되지 않는 메모리가 구현되므로 인해 메모리 손실이 발생한다. 또한 충분한 메모리를 제공받는다고 가정하더라도, 이를 전송하는 방식의 구현이 매우 어려우며, 수신기 쪽에서도 수신된 심볼의 위치를 정확히 파악하기가 매우 어렵다. 또한, 여러 종류의 인터리버가 사용된다는 면에서 각각 서로 다른 인터리버 방식(rule)을 메모리에 저장해두어야 하며, 이 때문에 제어기(cpu) 쪽에서도 상당히 많은 메모리 공간을 확보해야 하는 문제점이 있다.

상술한 바와 같이, 종래의 인터리빙 방식의 문제점은 하기와 같이 요약된다.

첫째, 기존의 인터리빙 방식으로는 인터리버의 크기가 2의 거듭제곱 형태로 표현될수 없고, 크기가 클수록 메모리 활용면에서 매우 비효율적이라는 문제점을 가진다. 즉, IMT-2000 순방향 링크를 위한 인터리버 설계에서 사실 각 논리 채널의 인터리버 크기가 2^m 꼴로 표현되지 않으며, 인터리버의 크기 또한 매우 크다는 점을 고려할 때 비트역상순 인터리빙 방식을 사용할수 없다.

둘째, 기존의 인터리빙 방식으로는 각각의 인터리버 크기에 따른 인터리빙 방식을 송수신기의 제어기(cpu=host)가 저장해두어야 하므로, 인터리버 버퍼 이외에 별도의 저장공간이 호스트(host) 메모리 쪽에 필요하다.

셋째, 송수신을 위한 인터리버/디인터리버 전송방식이 매우 복잡하며, 구현시 심볼동기를 맞추기가 어렵다.

발명이 이루고자 하는 기술적 과제

따라서 본 발명의 목적은 통신시스템에서 다양한 인터리버 크기에 대해 어드레스 번지를 하나의 알고리즘을 통해 생성하는 인터리빙 장치 및 방법을 제공함에 있다.

본 발명의 다른 목적은 통신시스템에서 인터리버 메모리가 프레임 크기 N만큼만 소요될수 있는 인터리빙 장치 및 방법을 제공함에 있다.

상술한 목적들을 달성하기 위한 통신시스템에서 인터리버 메모리에 순차적으로 저장된 데이타를 인터리빙하여 출력하기 위한 방법이, 인터리버 크기를 이진형태로 변환하여 최하위 비트로부터 최상위 비트 방향으로 연속되는 '0'비트수에 해당하는 영계수값을 구하는 단계와, 상기 '0'비트들을 제외한 비트들을 십진수로 변환하여 상한값을 구하는 단계와, 출력 순서번호를 상기 상한값으로 나누어 나머지값에 해당하는 제1변수값과 몫에 해당하는 제2변수값을 구하는 단계와, 상기 몫을 이진 형태로 변환한 후 비트역상순하여 십진형태로 변환하여 제3변수값을 구하는 단계와, 상기 나머지값과 2를 상기 영계수값으로 제곱근한 값을 곱한 후 이를 상기 제3변수값과 더해 인터리빙 어드레스 번지값을 생성하는 단계로 이루어져, 심볼 데이터가 순차적으로 저장된 인터리버 메모리에서 상기 생성된 인터리빙 어드레스 번지값에 대응되는 데이터를 출력함을 특징으로 한다.

발명의 구성 및 작용

이하 본 발명의 바람직한 실시예를 첨부된 도면의 참조와 함께 상세히 설명한다.

우선 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 한해서는 비록 다른 도면상에 표시되더라도 가능한 동일 부호를 가지도록 하였다. 또한 본 발명을 설명함에 있어서, 관련된 공지기능 혹은 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단된 경우 그 상세한 설명은 생략한다.

본 발명에서 사용하는 인터리버/디인터리버는 인터리빙/디인터리빙 알고리즘을 사용하여 입력으로 들어오는 심볼들이 순서를 치환한 뒤 출력버퍼에 새로운 순서에 따라 저장한다. 따라서 본 발명에서 제안한 인터리버/디인터리버는 3가지 부분(인터리버 메모리(입력 데이터 버퍼/출력 데이터 버퍼), 어드레스 생성부, 기존에 사용된 카운터)으로 구성된다.

도 2는 본 발명의 실시 예에 따른 인터리버의 구성도를 도시한 도면이다.

상기 도 2를 참조하면, 어드레스 생성부(211)는 인터리버 크기N, BIT_SHIFT, UP_LIMIT 값 및 클럭(clock)을 입력하여 읽기모드를 수행하기 위한 인터리버 메모리 어드레스를 생성하여 디인터리버 메모리(212)로 출력한다. 상기 인터리버 메모리(212)는 쓰기모드시 입력데이타를 순서대로 저장하고, 읽기모드시 상기 어드레스 생성부(211)에서 제공되는 어드레스에 따라 데이터를 출력한다. 카운터(213)는 클럭(Clock)을 입력하며, 상기 클럭을 카운터한 값을 상기 인터리버 메모리(212)의 쓰기 어드레스(Write) 값으로 출력한다.

상술한 바와 같이 상기 인터리버는 읽기모드시 입력데이타를 인터리버 메모리(212)에 순서대로 저장하고, 쓰기모드시 상기 어드레스 생성부(211)에서 발생된 읽기 어드레스에 따라 상기 인터리버 메모리(212)에 저장되어 있는 데이터를 출력한다.

여기서 상기 어드레스 생성부는 하기 수학식 1과 같은 부분분할 비트역상순 인터리빙 알고리즘에 따라 읽기 어드레스(=인터리빙 어드레스 번지값)를 생성한다.

$$For\ a\ given\ k\ .....(0 < k < N\text{-}1)$$

PLC = k mod UP_LIMIT;

PUC = k div UP_LIMIT;

AUC = PLC;

ALC = REVERSE_ORDER (PUC);

$$ADDRESS\_READ = AUC \times 2^{BIT\_SHIFT} + ALC$$

여기서 상기 k는 출력되는 데이터의 순서를 나타내는 것으로 순서번호라 칭하고, 상기 BIT_SHIFT는 LSB로부터 MSB방향으로 연속되는 '0'의 비트수를 나타내는 것으로 영계수값이라 칭하며, 상기 UP_LIMIT는 상기 연속되는 0를 제외한 비트들의 십진수에 해당하는 값으로 상한값이라 칭한다.

상기 수학식 1를 참조하여 쓰기 어드레스를 생성하는 방법을 설명하면, 우선 인터리버의 크기를 N이라 하자. 하기 수학식 1에서 K(=0,1,2,.....,N-1)는 입력데이타의 시간 인덱스(time index)를 나타내며, PLC와 PUC, ALC 및 AUC는 임의의 변수를 나타낸다. mod와 div는 각각 나머지와 몫을 구하는 모듈로 오퍼레이션(Modulo operation)과 디바이더 오퍼레이션(Divider operation)을 나타낸다. 또한, REVERSE_ORDER(H)는 H를 이진 포맷(binary format)으로 전환한 뒤, MSB로부터 LSB의 순서를 내림차순(reverse ordering)으로 하여 십진형태로 변환하는 비트역상순 함수(function)이다. 따라서 하기 수학식 1과 같은 함수를 이용하여 해당 입력데이타 시퀀스의 시간 인덱스 k에 대응되는 인터리버된 데이터의 읽기 시퀀스 인덱스인 ADDRESS_READ를 구하고, 해당되는 어드레스의 메모리 내용을 읽어내면 된다. 상기 UP_LIMIT와 BIT_SHIFT는 인터리버 크기에 의해서 결정되는 값이다. 일단, 인터리버 크기 N과 UP_LIMIT과 BIT_SHIFT가 결정되면, 이를 가지고 하기의 알고리즘에 따라 각각의 k에 해당되는 새로운 어드레싱 인덱스인 ADDRESS_READ를 생성하고, 이를 이용하여 메모리에 있는 인터리버 메모리212에서 데이터를 읽어간다.

상기 프레임크기(또는 인터리버 크기) N으로부터 UP_LIMIT와 BIT_SHIFT를 결정하는 방식을 설명하면, 임의의 인터리버 크기 N에 대하여 N-1을 이진수 형태로 표시한다. 그리고 LSB로부터 MSB방향으로 연속되는 '1'의 최대수를 구하고, 이를 BIT_SHIFT로 정의한다. 또한 이때의 위치를 j라 한다. 이후, MSB로부터 j+1번째 비트까지의 비트들(Truncated bits)을 모아서 십진수로 전환하여 UP_LIMIT로 정의한다.

예를 들어, N=576인 경우 이를 이진 형태로 나타내면 N=[10 0100 0000]이므로 BIT_SHIFT=6, j=6 따라서 $UP_LIMIT=(1001)_2=9$가 된다.

상술한 인터리버의 역에 해당하는 디인터리버의 구성은 도 3에 도시되어 있다.

상기 도 3을 참조하여 디인터리버의 구성을 살펴보면, 어드레스 생성부(311)는 인터리버 크기N, BIT_SHIFT, UP_LIMIT 및 클럭을 입력하여 쓰기모드를 수행하기 위한 인터리버 메모리 어드레스를 생성하여 디인터리버 메모리(312)로 출력한다. 상기 디인터리버 메모리(312)는 쓰기모드시 상기 어드레스 생성부(311)에서 제공되는 쓰기 어드레스(write ADDR)에 따라 입력데이타를 저장하고, 읽기모드시 저장 데이터를 순서대로 출력한다. 카운터(313)는 클럭(Clock)을 입력하며, 상기 클럭을 카운터한 값을 상기 디인터리버 메모리(312)의 읽기 어드레스(Read ADDR) 값으로 출력한다.

상기 디인터리버는 상기 인터리버의 역과정을 수행하는 것으로 구조상 모든 구성이 동일하며, 단지 쓰기모드시 상술한 수학식 1과 같은 알고리즘을 이용하여 입력데이터를 디인터리버 메모리(312)에 순서대로 저장하고, 읽기모드시 데이터를 순서대로 읽어간다는 점에서 다르다. 즉, 상기 디인터리버는 송신측에서 보낸 데이터를 원래의 순서대로 복원하기 위해서 저장시 미리 데이터의 원 순서를 찾아 저장하는 것이다.

따라서, 이하 설명은 인터리버 위주로 설명할 것이다.

그러면, 본 발명을 차세대 이동통신시스템인 IMT2000(CDMA2000) 시스템에 적용할 경우의 실제적인 예들을 살펴본다.

우선, IMT2000 시스템의 순방향 링크에서 사용되는 인터리버의 크기를 표 1를 통해 살펴본다.

[표 1]

	순방향 기본채널 (RS1)	순방향 기본채널 (RS2)	순방향 부가채널 (RS1)	순방향 부가채널 (RS2)	순방향 공통제어채널	순방향 동기채널	순방향 페이징채널	순방향 전용제어채널
72(bit)								
144	○ (5msec)	○ (5msec)						○ (5msec)
192						○ (26.6msec)		
288								
384								

576	○	○	○	○	○		○	○ (20msec)
1152		○	○	○				
2304			○	○				
4608			○	○				
9216			○	○				
18432			○	○				
36864			○	○				

상기 표 1에서와 같이 IMT2000 시스템에서는 12개의 인터리버 크기N가 제안되어 있으며, 이는 각 순방향 논리채널에 적정히 사용된다. 여기서 각 순방향 채널에서 사용되는 인터리버들은 '○'로 표시하였다. 일예를 들어, 순방향 기본채널(제2레이트)인 경우, 사용되는 인터리버 크기는 144bit(이때의 프레임 사이즈는 5msec이다), 576bit 및 1152bit이다.

여기서 상기 표 1에서 제시된 인터리버 크기에 해당하는 상기한 UP_LIMIT와 BIT_SHIFT 값을 구해보면 하기 표 2와 같다.

[표 2]

인터리버 크기(N)	N의 이진형태	UP_LIMI	BIT_SHIFT	논리 채널
144	10010	9(1001)	4	5msec/frame 순방향 전용제어채널(5msec/frame) 순방향 기본채널/RS2(5msec/frame)
192	1100000	3(0110)	5	동기채널(26.22msec/frame)
576	1001000000	9(1001)	6	순방향 페이징채널, 순방향 공통제어채널 순방향 전용제어채널(20msec/frame) 순방향 기본채널/RS2 순방향 부가채널/RS1
1152	10010000000	9(1001)	7	순방향 기본채널(RS2) 순방향 부가채널
2304	100100000000	9(1001)	8	순방향 부가채널
4608	1001000000000	9(1001)	9	순방향 부가채널
9216	10010000000000	9(1001)	10	순방향 부가채널
18432	100100000000000	9(1001)	11	순방향 부가채널
36864	1001000000000000	9(1001)	12	순방향 부가채널

상기 표 2를 참조하여 인터리버 크기N이 9216인 경우 BIT_SHIFT와 UP_LIMIT를 구하는 방법을 설명하면, 우선 상기 9216을 이진형태로 나타내면 N=[10 0100 0000 0000]이다. 여기서 LSB로부터 MSB방향으로 연속되는 '0'의 최대수를 구하고, 이를 BIT_SHIFT로 정의한다. 또한 이때의 위치를 j라고 한다. 따라서 상기 BIT_SHIFT는 10이 되고, 상기 j 값 또한 10이 된다. 그리고 상기 MSB로부터 j+1번째까지의 비트들을 모아서 이를 십진수($1001=9_{(10)}$)를 전환하여 UP_LIMIT(9)를 구한다.

하기 표 3(쓰기모드) 및 표 4(읽기모드)는 N=576인 인터리버에 대하여 읽기-모드(Read-Mode)와 쓰기-모드(Write-Mode)의 일예를 보여준다.

[표 3]

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
541	542	543	544	545	546	547	548	549	550
551	552	553	554	555	556	557	558	559	560
561	562	563	564	565	566	567	568	569	570
571	572	573	574	575	576				

[표 4]

1	65	129	193	257	321	385	449	513
33	97	161	225	289	353	417	481	545
17	81	145	209	273	337	401	465	529
49	113	177	241	305	369	433	497	561
9	73	137	201	265	329	393	457	521
41	105	169	233	297	361	425	489	553
25	89	153	217	281	345	409	473	537
57	121	185	249	313	377	441	505	569
5	69	133	197	261	325	389	453	517
…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…
16	80	144	208	272	336	400	464	528
48	112	176	240	304	368	432	496	560
32	96	160	224	288	352	416	480	544
64	128	192	256	320	384	448	512	576

쓰기-모드에서는 상기 표 1에서와 같이 000번지로터 574번지까지 순차적으로 입력데이타를 인터리버 메모리에 저장한다. 다음으로, 읽기-모드에서는 어드레싱 생성부211에서 생성되는 Read ADDR를 사용하여 해당하는 번지의 데이터를 인터리버 메모리로부터 출력한다.

예를 들어, 3번째(k=2)로 출력될 데이터가 어떤것인지를 상기한 수학식 1를 통해 살펴보자. 우선 N(576)으로부터 BIT_SHIFT와 UP_LIMT를 구해보면, BIT_SHIFT는 6이고, UP_LIMIT는 9이다. 따라서 PLC = 2 mod 9 = 2이고, PUC = 2 div 9 = 0이다. 여기서 상기 PLC 값은 AUC 값으로 대치된다. 또한, ALC = REVERSE_ORDER(2) = 1이다. 따라서 최종적으로 구해지는 어드레스 번지 ADDRESS_READ = 2×2^{6} + 1 = 129이다.

발명의 효과

상술한 바와 같이 본 발명은 2의 거듭제곱으로 표현되지 않는 다양한 인터리버 크기에 대하여 효과적인

어드레스 번지 생성방법을 제안하였다. 따라서 기존에 2의 거듭제곱으로 표현되지 않는 인터리버가 메모리의 사용면에서 비효율적인 것을 해결하였다. 그리고 다양한 인터리버 사이즈에 대해 어드레스 번지를 하나의 알고리즘을 통해 생성할수 있으므로, 기존에 호스트(CPU)가 각 인터리버 사이즈에 대해 인터리빙 방식을 저장하므로서 소비되는 메모리 공간을 제거하였다. 또한 본 발명은 프레임 크기인 N비트만큼의 인터리버 메모리가 소요되므로 메모리 활용을 증대시켰다.

(57) 청구의 범위

청구항 1

통신시스템의 인터리빙 장치에 있어서,

인터리버 크기 및 상기 인터리버 크기에 따라 구해진 변수들을 입력하여 인터리빙 어드레스 번지값을 출력하는 어드레스 생성부와,

입력데이타를 순차적으로 저장하고, 상기 어드레스 생성부에서 제공되는 어드레스 번지값에 해당하는 데이터를 출력하는 인터리빙 메모리로 구성됨을 특징으로 하는 인터리빙 장치.

청구항 2

제1항에 있어서,

상기 인터리버 크기에 따라 구해진 변수들이,

상기 인터리버 크기를 이진형태로 나타내었을 때 최하위 비트로부터 연속되는 '0'비트의 수를 나타내는 영계수값과,

상기 연속되는 '0'비트들을 제외한 비트들의 십진수에 해당하는 상한값으로 이루어짐을 특징으로 하는 인터리빙 장치.

청구항 3

제2항에 있어서,

상기 어드레스 생성부가 하기 수학식 2에 의해서 인터리빙 어드레스 번지값을 생성함을 특징으로 하는 인터리빙 장치.

$$For\ a\ given\ k\ .....(0 < k < N\text{-}1)$$

PLC = k mod UP_LIMIT;

PUC = k div UP_LIMIT;

AUC = PLC;

ALC = REVERSE_ORDER (PUC);

$$ADDRESS\_READ = AUC \times 2^{BIT\_SHIFT} + ALC;$$

여기서 상기 PLC, PUC, ALC 및 AUC는 임의의 변수를 나타내고, k(순서번호)는 출력되는 데이터의 순서를 나타내며, 상기 REVERCE_ORDER(PUC)는 PUC를 이진 형태로 변환한 후 최상위 비트로부터 최하위 비트의 순서를 내림차순으로 정열하여 십진형태로 변환하는 비트역상순 함수를 나타내고, 상기 ADDRESS_READ는 출력되는 데이터의 어드레스 번지값임.

청구항 4

통신시스템의 디인터리빙 장치에 있어서,

디인터리버 크기 및 상기 디인터리버 크기에 따라 구해진 변수들을 입력하여 디인터리빙 어드레스 번지값을 출력하는 어드레스 생성부와,

상기 어드레스 생성부에서 제공되는 어드레스 번지값에 입력 데이터를 저장하며, 상기 저장된 데이터를 순차적으로 출력하는 인터리빙 메모리로 구성됨을 특징으로 하는 디인터리빙 장치.

청구항 5

제4항에 있어서,

상기 디인터리버 크기에 따라 구해진 변수들이,

상기 디인터리버 크기를 이진형태로 나타내었을 때 최하위 비트로부터 연속되는 '0'비트의 수를 나타내는 영계수값과,

상기 연속되는 '0'비트들을 제외한 비트들의 십진수에 해당하는 상한값으로 이루어짐을 특징으로 하는 디인터리빙 장치.

청구항 6

제5항에 있어서,

상기 어드레스 생성부가 하기 수학식 3에 의해서 디인터리빙 어드레스 번지값을 생성함을 특징으로 하는 디인터리빙 장치.

$$For\ a\ given\ k\ .....(0 < k < N\text{-}1)$$

PLC = k mod UP_LIMIT;

PUC = k div UP_LIMIT;

AUC = PLC;

ALC = REVERSE_ORDER (PUC);

$$ADDRESS\_READ=AUC\times 2^{BIT\_SHIFT}+ALC;$$

여기서 상기 PLC, PUC, ALC 및 AUC는 임의의 변수를 나타내고, k(순서번호)는 입력되는 데이터의 순서를 나타내며, 상기 REVERCE_ORDER(PUC)는 PUC를 이진 형태로 변환한 후최상위 비트로부터 최하위 비트의 순서를 내림차순으로 정열하여 십진형태로 변환하는 비트역상순 함수를 나타내고, 상기 ADDRESS_READ는 출력되는 데이터의 어드레스 번지값임.

청구항 7

통신시스템에서 인터리버 메모리에 순차적으로 저장된 데이타를 인터리빙하여 출력하기 위한 방법에 있어서,

인터리버 크기를 이진형태로 변환하여 최하위 비트로부터 최상위 비트 방향으로 연속되는 '0'비트수에 해당하는 영계수값을 구하는 단계와,

상기 '0'비트들을 제외한 비트들을 십진수로 변환하여 상한값을 구하는 단계와,

출력 순서번호를 상기 상한값으로 나누어 나머지값에 해당하는 제1변수값과 몫에 해당하는 제2변수값을 구하는 단계와,

상기 상기 제2변수값을 이진 형태로 변환한 후 비트역상순하여 십진형태로 변환하여 제3변수값을 구하는 단계와,

상기 나머지값과 2를 상기 영계수값으로 제곱근한 값을 곱한 후 이를 상기 제3변수값과 더해 인터리빙 어드레스 번지값을 생성하는 단계로 이루어져,

심볼 데이터가 순차적으로 저장된 인터리버 메모리에서 상기 생성된 인터리빙 어드레스 번지값에 대응되는 데이터를 출력함을 특징으로 하는 인터리빙 방법.

청구항 8

통신시스템의 인터리빙 방법에 있어서,

입력데이타를 인터리버 메모리에 순차적으로 저장하는 과정과,

저장완료시 상기 인터리버 메모리 크기 및 상기 크기를 이용하여 산출된 각종 변수들을 가지고 인터리빙 어드레스 번지값을 순차적으로 생성하는 과정과,

상기 생성된 어드레스 번지값에 대응되는 데이터를 출력하는 과정으로 이루어짐을 특징으로 하는 인터리빙 방법.

청구항 9

제8항에 있어서,

상기 인터리버 메모리 크기를 이용하여 산출된 변수들이,

상기 인터리버 크기를 이진형태로 나타내었을 때 최하위 비트로부터 연속되는 '0'비트의 수를 나타내는 영계수값과,

상기 연속되는 '0'비트들을 제외한 비트들의 십진수에 해당하는 상한값으로 이루어짐을 특징으로 하는 인터리빙 방법.

청구항 10

제8항에 있어서,

상기 인터리빙 어드레스 번지값이 하기 수학식 4에 의해 생성됨을 특징으로 하는 인터리빙 방법.

$$For\ a\ given\ k\ .....(0 < k < N\text{-}1)$$

PLC = k mod UP_LIMIT;

PUC = k div UP_LIMIT;

AUC = PLC;

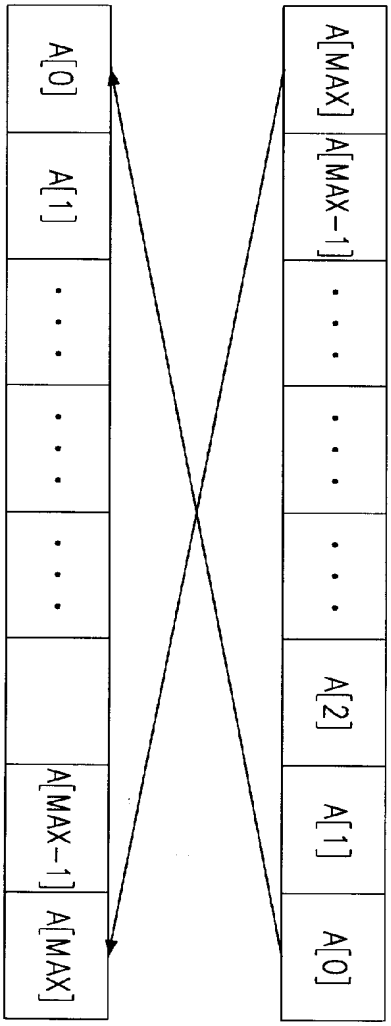
ALC = REVERSE_ORDER (PUC);

$$ADDRESS\_READ = AUC \times 2^{BIT\_SHIFT} + ALC;$$

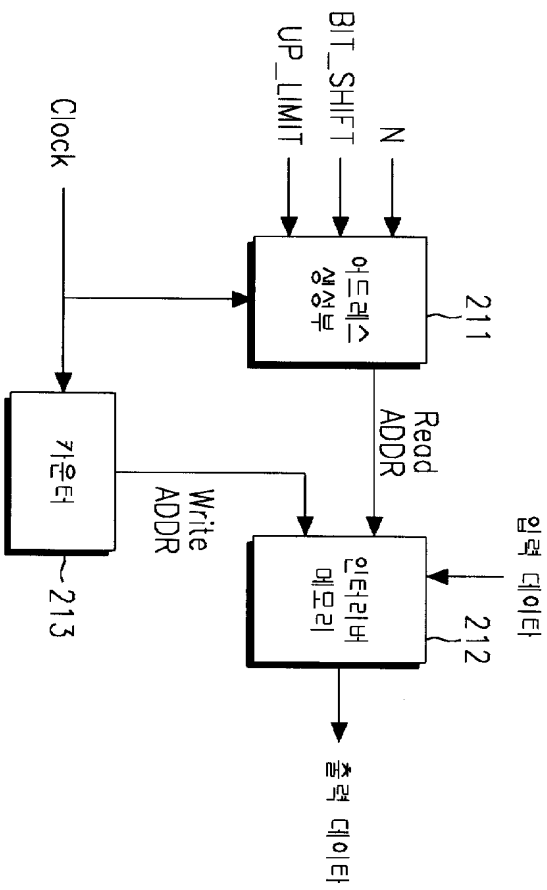
여기서 상기 PLC, PUC, ALC 및 AUC는 임의의 변수를 나타내고, k는 입력되는 데이터의 순서를 나타내는 것으로 순서번호이며, 상기 REVERCE_ORDER(PUC)는 PUC를 이진 형태로 변환한 후 최상위 비트로부터 최하위 비트의 순서를 내림차순으로 정열하여 십진형태로 변환하는 비트역상순 함수를 나타내고, 상기 ADDRESS_READ는 출력되는 데이터의 어드레스 번지값임.

도면

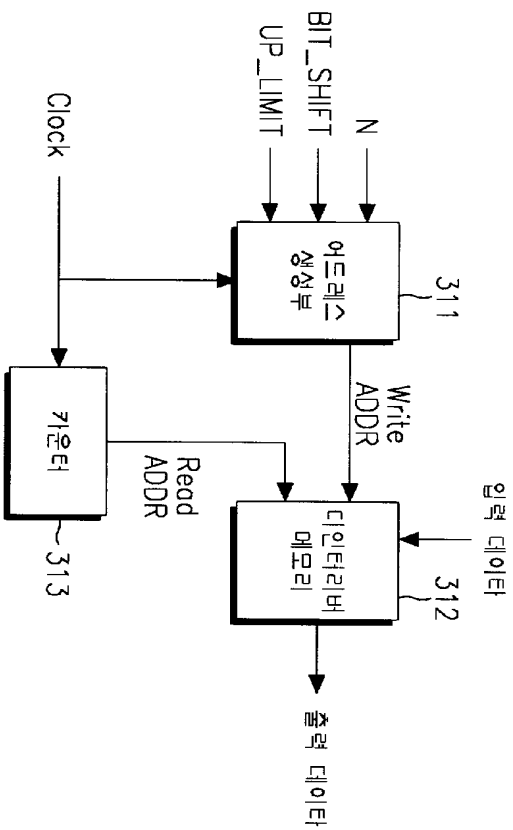
도면1



도면2


Clock
UP_LIMIT
BIT_SHIFT
N
어드레스 생성부
211
Read ADDR
입력 데이터
인터리버 메모리
212
출력 데이터
Write ADDR
카운터
213

도면3


Clock
UP_LIMIT
BIT_SHIFT
N
어드레스 생성부
311
Write ADDR
Read ADDR
카운터
313
디인터리버 메모리
312
입력 데이타
출력 데이타